AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (currently amended) A computation method ~~of~~ for simulating behavior of a flow interacting with an object, comprising the steps of:

~~the method providing~~

developing a simulated numerical representation in N dimensions, $N \geq 3$, ~~that is~~ composed of a plurality of approximated values at a multitude of points in at least a part of space where ~~the~~ a flow interacts with ~~the~~ an object, the approximated values being of a physical parameter u of the flow to which is associated a velocity field $\vec{a}$ which determines a preferential direction [[,]] by ~~means of~~ executing a numerical scheme wherein at least one spatial p^{th} derivative D_p, $p \geq 1$, of the parameter u is approximated at the points of the part of space, through ~~the method comprising~~ the steps of~~:~~

i) constructing a discrete N-dimensional grid constructed by N families of coordinate lines and using the constructed grid ~~using~~ for the part of space, ~~a discrete N-dimensional grid constructed by N families of coordinate lines;~~

ii) computing, in at least one point P of the grid, called the point of computation, an approximated value D_p^A of D_p with an error ε_n, by using values u_s of the parameter in a collection of grid points, called the stencil S, and computational functions, evaluated with the values u_s, the computational functions depending on the numerical framework in which D_p is expressed, and

iii) choosing the computational functions for the approximated value D_p^A in such a way that the approximated value D_p^A is optimized for the preferential direction, ~~and~~

wherein the stencil S contains at least one point situated outside all the coordinate lines passing through the point of computation P, and the stencil S contains at least a first point and a second point, the first point being defined by N first coordinate lines of the N families of lines, the second point being defined by N second coordinate lines of the N families of lines, and for at least one family N_f of the coordinate lines, the first coordinate line belonging to the family N_f is different from and not adjacent to the second coordinate line belonging to the same family N_f; ~~and~~

outputting the simulated numerical representation that simulates, for the part of space, the behavior of the flow interacting with the object; and

determining the behavior of the flow interacting with the object by using the output simulated numerical representation.

2. (original) The method of claim 1, wherein for varying preferential directions, the approximation D_p^A depends continuously on the values u_s.

3. (original) The method of claim 1, wherein the computing step of D_p^A comprises the steps of:

providing a local basis $B(\vec{e}_1,\vec{e}_2,\vec{e}_3,\ldots)$ of curvilinear coordinates which has the unit vector $\vec{e}_1$ along the preferential direction, and

choosing the computational functions so that a contribution to the error ε_n of at least one pure or one mixed derivative as expressed in the local basis B is minimized, while using as a formulation of the values u_s of the parameter at each of the points of the stencil S, a truncated Taylor series expansion with respect to the point of computation P with an error, called the truncation error ε_s.

4. (original) The method of claim 3, wherein the computational functions are individual coefficients C_S and the approximated value D_p^A is a linear combination of values u_s of the parameter.

5. (original) The method of claim 3, wherein the computing step of D_p^A comprises the steps of:

using an integral formulation for the computation of the derivative D_p with computational functions which are fluxes through a control volume,

computing in at least one volume an approximated value D_p^A of D_p with an error ε_n, where the approximated value is a function of the flux formulation employed.

6. (original) The method of claim 3, wherein the computing step of D_p^A comprises the steps of:

subdividing the part of space into elements, containing nodes at which the approximation of the physical parameter u is stored,

defining basis computational functions φ, called interpolation functions, on the elements, where the interpolation

functions are used to approximate the physical value u on the element,

computing the integral of the derivative D_p on the element, using a test computational function ψ, called weighting function,

computing in at least one element an approximated value $D_p^{\varphi,\psi}$ of D_p with an error ε_n, where the approximated value is a function of the interpolation functions φ and the weighting function ψ.

7. (original) The method of claim 3, wherein the computing step of D_p^A comprises the steps of:

subdividing the part of space into elements, containing nodes at which the approximation of the physical parameter u is stored,

defining basis computational functions φ, called interpolation functions, on the elements, where the interpolation functions are used to approximate the physical value u on the element,

computing the integral I_{el} of the derivative D_p on volumes, and distributing parts $\alpha_i I_{el}$ to nodes i, where α_i represent computational functions called distribution coefficients.

8. (currently amended) The method of claim 3, wherein the computing step of D_p^A comprises the steps of:

subdividing the part of space into elements, containing nodes at which the approximation of the physical parameter u is stored,

defining basis computational functions φ, called interpolation functions, on the elements, where the interpolation functions are used to approximate the physical value u on the element,

computing fluxes f at the surfaces of volumes, and distributing parts $\alpha_i f$ to nodes i, where α_i represent computational functions called distribution coefficients.
[[,]]

9. (original) The method of claim 1, wherein the computing step of D_p^A comprises the steps of:

using a representation of the numerical solutions in Fourier components, and

choosing in the approximated value D_p^A the computational functions in such a way that the Fourier components are optimized for certain directions which are related to the

velocity $\vec{a}$, while using the values u_s of the parameter at each of the points of the stencil S in the Fourier representation.

10. (original) The method of claim 9, wherein the computational functions are individual coefficients C_S and the approximated value D_p^A is a linear combination of values u_s of the parameter.

11. (original) The method of claim 9, wherein the computing step of D_p^A comprises the steps of:

using an integral formulation for the computation of the derivative D_p with computational functions which are fluxes through a control volume,

computing in at least one volume an approximated value D_p^A of D_p with an error ε_n, where the approximated value is a function of the flux formulation employed.

12. (original) The method of claim 9, wherein the computing step of D_p^A comprises the steps of:

subdividing the part of space into elements, containing nodes at which the approximation of the physical parameter u is stored,

defining basis computational functions φ, called interpolation functions, on the elements, where the interpolation functions are used to approximate the physical value u on the element,

computing the integral of the derivative D_p on the element, using a test computational function ψ, called weighting function,

computing in at least one element an approximated value $D_p^{\varphi,\psi}$ of D_p with an error ε_n, where the approximated value is a function of the interpolation functions φ and the weighting function ψ.

13. (original) The method of claim 9, wherein the computing step of D_p^A comprises the steps of:

subdividing the part of space into elements, containing nodes at which the approximation of the physical parameter u is stored,

defining basis computational functions φ, called interpolation functions, on the elements, where the interpolation functions are used to approximate the physical value u on the element,

computing the integral I_{el} of the derivative D_p on volumes, and distributing parts $\alpha_i I_{el}$ to nodes i, where α_i

represent computational functions called distribution coefficients.

14. (currently amended) The method of claim 9, wherein the computing step of D_p^A comprises the steps of:

subdividing the part of space into elements, containing nodes at which the approximation of the physical parameter u is stored,

defining basis computational functions φ, called interpolation functions, on the elements, where the interpolation functions are used to approximate the physical value u on the element,

computing fluxes f at the surfaces of volumes, and distributing parts $\alpha_i f$ to nodes i, where α_i represent computational functions called distribution coefficients. [[,]]

15. (original) The method of claim 1, wherein the N-dimensional grid is expressed in a coordinate system which is chosen from the group consisting of: rectangular coordinates, spherical coordinates, cylindrical coordinates, parabolic cylindrical coordinates, paraboloidal coordinates, elliptic cylindrical coordinates, prolate spheroidal coordinates, oblate spheroidal coordinates, bipolar coordinates, toroidal

coordinates, conical coordinates, confocal ellipsoidal coordinates and confocal paraboloidal coordinates.

16. (original) The method of claim 1, wherein the N-dimensional grid is chosen from the group consisting of: a grid with non-uniform mesh spacing, a grid which is moving, a grid which is deforming, a grid which is rotating, and a grid which is staggered, and any combination thereof.

17. (original) The method of claim 1, wherein the spatial p^{th} derivative D_p is a pure derivative $\frac{\partial^p u}{\partial e_i^p}$.

18. (original) The method of claim 1, wherein $p=1$.

19. (original) The method of claim 1, wherein $p=2$.

20. (original) The method of claim 1, wherein the spatial p^{th} derivative D_p is a mixed derivative $\frac{\partial^p u}{\partial e_1^{p_1} \partial e_2^{p_2} \ldots}$ with $p_1 + p_2 + \ldots = p$.

21. (original) The method of claim 3, computing an approximated value D_1^A of D_1, where the approximated value is denoted by

$$\begin{pmatrix} (u_x)_{i,j,k,\ldots} \\ (u_y)_{i,j,k,\ldots} \\ (u_z)_{i,j,k,\ldots} \\ \vdots \end{pmatrix}, \qquad (27)$$

wherein

$$(u_x)_{i,j,k,\ldots} = \frac{1}{\Delta x}\Big\{a_{-m}u_{i-m,j,k,\ldots} + a_{-m+1}u_{i-m+1,j,k,\ldots} + \cdots + a_{n-1}u_{i+n-1,j,k,\ldots} + a_n u_{i+n,j,k,\ldots} + T_x\Big\},$$

$$\begin{aligned}(u_y)_{i,j,k,\ldots} = \frac{1}{\Delta y}\Big\{&a_{-m}(u_{i-m,j-m,k,\ldots} - u_{i-m,j,k,\ldots}) \\ &+ a_{-m+1}(u_{i-m+1,j-m+1,k,\ldots} - u_{i-m+1,j,k,\ldots}) + \cdots \\ &+ a_{n-1}(u_{i+n-1,j+n-1,k,\ldots} - u_{i+n-1,j,k,\ldots}) \\ &+ a_n(u_{i+n,j+n,k,\ldots} - u_{i+n,j,k,\ldots}) + T_y\Big\},\end{aligned}$$

$$\begin{aligned}(u_z)_{i,j,k,\ldots} = \frac{1}{\Delta z}\Big\{&a_{-m}(u_{i-m,j-m,k-m,\ldots} - u_{i-m,j-m,k,\ldots}) \\ &+ a_{-m+1}(u_{i-m+1,j-m+1,k-m+1,\ldots} - u_{i-m+1,j-m+1,k,\ldots}) + \cdots \\ &+ a_{n-1}(u_{i+n-1,j+n-1,k+n-1,\ldots} - u_{i+n-1,j+n-1,k,\ldots}) \\ &+ a_n(u_{i+n,j+n,k+n,\ldots} - u_{i+n,j+n,k,\ldots}) + T_z\Big\},\end{aligned}$$

$$\vdots \qquad (28)$$

wherein $a_{-m} \neq 0$, $a_n \neq 0$, m and n are given integers, $m+n>0$, and $m+n>1$ if $m*n=0$, the terms $T_x, T_y, T_z, \ldots$ represent the degrees of freedom which are used in the optimization of the

approximated value D_1^A, and where indices $(i,j,k,...)$ define the point of computation P on the N-dimensional grid, and $\Delta x, \Delta y, \Delta z, ...$ denote the mesh spacings of the N-dimensional grid in each coordinate direction.

22. (original) The method of claim 21, computing an approximated value D_1^A of $D_1 = \frac{\partial u}{\partial e_1}$ by a discretization of order M, in which the terms $\frac{\partial^{M+1} u}{\partial e_2^{M_2} \partial e_3^{M_3} \ldots}$ with $M_2 + M_3 + \ldots = M + 1$ are eliminated in the approximated value D_1^A.

23. (original) The method of claim 21, computing an approximated value D_1^A of $D_1 = \frac{\partial u}{\partial e_1}$ by a discretization of order M, in which the terms $\frac{\partial^{M+1} u}{\partial e_1^{M_1} \partial e_2^{M_2} \partial e_3^{M_3} \ldots}$ with $M_1 + M_2 + M_3 + \ldots = M + 1$ and $M_1 < M + 1$ are eliminated in the approximated value D_1^A in the case that $\vec{e}_1$ is along the x-axis or along diagonals.

24. (original) The method of claim 1, computing an approximated value D_1^A of D_1, where the approximated value is denoted by

$$\begin{pmatrix} (u_x)_{i,j,k,\ldots} \\ (u_y)_{i,j,k,\ldots} \\ (u_z)_{i,j,k,\ldots} \\ \vdots \end{pmatrix}, \qquad (29)$$

wherein

$$(u_x)_{i,j,k,\ldots} = \frac{1}{\Delta x}\Big\{a_{-m}u_{i-m,j,k,\ldots} + a_{-m+1}u_{i-m+1,j,k,\ldots} + \cdots + a_{n-1}u_{i+n-1,j,k,\ldots} + a_n u_{i+n,j,k,\ldots}\Big\},$$

$$\begin{aligned}(u_y)_{i,j,k,\ldots} = \frac{1}{\Delta y}\Big\{ & a_{-m}(u_{i-m,j-m,k,\ldots} - u_{i-m,j,k,\ldots}) \\ & + a_{-m+1}(u_{i-m+1,j-m+1,k,\ldots} - u_{i-m+1,j,k,\ldots}) + \cdots \\ & + a_{n-1}(u_{i+n-1,j+n-1,k,\ldots} - u_{i+n-1,j,k,\ldots}) \\ & + a_n(u_{i+n,j+n,k,\ldots} - u_{i+n,j,k,\ldots})\Big\},\end{aligned}$$

$$\begin{aligned}(u_z)_{i,j,k,\ldots} = \frac{1}{\Delta z}\Big\{ & a_{-m}(u_{i-m,j-m,k-m,\ldots} - u_{i-m,j-m,k,\ldots}) \\ & + a_{-m+1}(u_{i-m+1,j-m+1,k-m+1,\ldots} - u_{i-m+1,j-m+1,k,\ldots}) + \cdots \\ & + a_{n-1}(u_{i+n-1,j+n-1,k+n-1,\ldots} - u_{i+n-1,j+n-1,k,\ldots}) \\ & + a_n(u_{i+n,j+n,k+n,\ldots} - u_{i+n,j+n,k,\ldots})\Big\},\end{aligned}$$

$$\vdots \qquad (30)$$

wherein $a_{-m} \neq 0$, $a_n \neq 0$, m and n are given integers, $m+n>0$, and $m+n>1$ if $m*n=0$, and where indices $(i,j,k,\ldots)$ define the point of computation P on the N-dimensional grid, and $\Delta x, \Delta y, \Delta z, \ldots$ denote the mesh spacings of the N-dimensional grid in each coordinate direction.

25. (original) The method of claim 21, wherein

$$(u_x)_{i,j,k,\dots} = \frac{1}{\Delta x}\left\{\frac{1}{2}(u_{i+1,j,k,\dots} - u_{i-1,j,k,\dots}) + T_x\right\},$$

$$(u_y)_{i,j,k,\dots} = \frac{1}{\Delta y}\left\{\frac{1}{2}(u_{i+1,j+1,k,\dots} - u_{i+1,j,k,\dots} + u_{i-1,j,k,\dots} - u_{i-1,j-1,k,\dots}) + T_y\right\},$$

$$(u_z)_{i,j,k,\dots} = \frac{1}{\Delta z}\left\{\frac{1}{2}(u_{i+1,j+1,k+1,\dots} - u_{i+1,j+1,k,\dots} + u_{i-1,j-1,k,\dots} - u_{i-1,j-1,k-1,\dots}) + T_z\right\},$$

$$\vdots \qquad (31)$$

.

26. (original) The method of claim 21, wherein

$$(u_x)_{i,j,k,\dots} = \frac{1}{\Delta x}\left\{\frac{1}{12}(u_{i-2,j,k,\dots} - 8u_{i-1,j,k,\dots} + 8u_{i+1,j,k,\dots} - u_{i+2,j,k,\dots}) + T_x\right\},$$

$$\begin{aligned}(u_y)_{i,j,k,\dots} = \frac{1}{\Delta y}\Big\{\frac{1}{12}(&u_{i-2,j-2,k,\dots} - u_{i-2,j,k,\dots} \\ &- 8u_{i-1,j-1,k,\dots} + 8u_{i-1,j,k,\dots} \\ &+ 8u_{i+1,j+1,k,\dots} - 8u_{i+1,j,k,\dots} \\ &- u_{i+2,j+2,k,\dots} + u_{i+2,j,k,\dots}) + T_y\Big\},\end{aligned}$$

$$\begin{aligned}(u_z)_{i,j,k,\dots} = \frac{1}{\Delta z}\Big\{\frac{1}{12}(&u_{i-2,j-2,k-2,\dots} - u_{i-2,j-2,k,\dots} \\ &- 8u_{i-1,j-1,k-1,\dots} + 8u_{i-1,j-1,k,\dots} \\ &+ 8u_{i+1,j+1,k+1,\dots} - 8u_{i+1,j+1,k,\dots} \\ &- u_{i+2,j+2,k+2,\dots} + u_{i+2,j+2,k,\dots}) + T_z\Big\},\end{aligned}$$

$$\vdots \qquad (32)$$

.

27. (original) The method of claim 21, wherein

$$(u_x)_{i,j,k,\ldots} = \frac{1}{\Delta x}\left\{\frac{3}{2}u_{i,j,k,\ldots} - 2u_{i-1,j,k,\ldots} + \frac{1}{2}u_{i-2,j,k,\ldots} + T_x\right\},$$

$$(u_y)_{i,j,k,\ldots} = \frac{1}{\Delta y}\left\{-2(u_{i-1,j-1,k,\ldots} - u_{i-1,j,k,\ldots}) + \frac{1}{2}(u_{i-2,j-2,k,\ldots} - u_{i-2,j,k,\ldots}) + T_y\right\},$$

$$(u_z)_{i,j,k,\ldots} = \frac{1}{\Delta z}\left\{-2(u_{i-1,j-1,k-1,\ldots} - u_{i-1,j-1,k,\ldots}) + \frac{1}{2}(u_{i-2,j-2,k-2,\ldots} - u_{i-2,j-2,k,\ldots}) + T_z\right\},$$

$$\vdots \qquad (33)$$

28. (currently amended) A ~~simulation~~ method according to claim 1, in three dimensions, obtaining the approximation D_p^A with order M of the derivative $D_p = \dfrac{\partial^p u}{\partial e_1^{p_1}\,\partial e_2^{p_2}\,\partial e_3^{p_3}}$ with $p_1 + p_2 + p_3 = p$ on a grid of given extent ~~from the output of the program generate-discretizations which is given in appendices 1-5~~ from the output of a computer program.

29. (currently amended) A ~~simulation~~ method according to claim 28, wherein ~~the input parameter optimize has the value 1 or 2~~ in the approximation D_p^A by a discretization of order M

terms are eliminated, which are the terms $\dfrac{\partial^{M+1}u}{\partial e_2^{M_2}\partial e_3^{M_3}\ldots}$ with $M_2+M_3+\ldots=M+1$, or the terms terms $\dfrac{\partial^{M+1}u}{\partial e_1^{M_1}\partial e_2^{M_2}\partial e_3^{M_3}\ldots}$ with $M_1+M_2+M_3+\ldots=M+1$ and $M_1<M+1$ in the case that $\vec{e}_1$ is along the x-axis or along diagonals.

30. (currently amended) A ~~simulation~~ method according to claim 28, wherein ~~the input parameter order has the value 1~~ the approximation of D_p^A is a first order accurate discretization.

31. (original) The method of claim 3, computing in three dimensions an approximation of the second derivative $D_2=\dfrac{\partial^2 u}{\partial e_1^2}$, where the approximation D_2^A is expressed in the terms $u_{xx}, u_{yy}, u_{zz}, u_{xy}, u_{yz}$ and u_{zx} which are given by

$$(u_{xx})_{i,j,k}=\frac{1}{(\Delta x)^2}(u_{i+1,j,k}-2u_{i,j,k}+u_{i-1,j,k}+T_{xx}),$$

$$(u_{yy})_{i,j,k}=\frac{1}{(\Delta y)^2}(u_{i,j+1,k}-2u_{i,j,k}+u_{i,j-1,k}+T_{yy}),$$

$$(u_{zz})_{i,j,k}=\frac{1}{(\Delta z)^2}(u_{i,j,k+1}-2u_{i,j,k}+u_{i,j,k-1}+T_{zz}),$$

$$(u_{xy})_{i,j,k} = \frac{1}{\Delta x \Delta y}\left\{\frac{1}{2}(u_{i+1,j+1,k} - u_{i,j+1,k} - u_{i+1,j,k} + 2u_{i,j,k} - u_{i-1,j,k} - u_{i,j-1,k} + u_{i-1,j-1,k}) + T_{xy}\right\},$$

$$(u_{yz})_{i,j,k} = \frac{1}{\Delta y \Delta z}\left\{\frac{1}{4}(u_{i+1,j+1,k+1} - u_{i+1,j,k+1} - u_{i+1,j+1,k} + u_{i+1,j,k} + u_{i,j+1,k+1} - u_{i,j,k+1} - u_{i,j+1,k} + 2u_{i,j,k} - u_{i,j-1,k} - u_{i,j,k-1} + u_{i,j-1,k-1} + u_{i-1,j-1,k-1} - u_{i-1,j-1,k} - u_{i-1,j,k-1} + u_{i-1,j,k}) + T_{yz}\right\},$$

$$(u_{zx})_{i,j,k} = \frac{1}{\Delta z \Delta x}\left\{\frac{1}{4}(u_{i+1,j+1,k+1} - u_{i,j+1,k+1} - u_{i+1,j+1,k} + u_{i,j+1,k} + u_{i+1,j,k+1} - u_{i+1,j,k} - u_{i,j,k+1} + 2u_{i,j,k} - u_{i-1,j,k} - u_{i,j,k-1} + u_{i-1,j,k-1} + u_{i-1,j-1,k-1} - u_{i-1,j-1,k} - u_{i,j-1,k-1} + u_{i,j-1,k}) + T_{zx}\right\},$$

(34)

wherein the terms $T_{xx}, T_{xy}, T_{xz}, T_{yy}, T_{yz}$ and T_{zz} represent the degrees of freedom which are used in the optimization of the approximated value D_2^A, and where indices (i,j,k) define the point of computation P on the three-dimensional grid, and $\Delta x, \Delta y, \Delta z$ denote the mesh spacings of the three-dimensional grid in each coordinate direction.

32. (original) The method according to claim 1, wherein

$D_p^A = \sum_n L_n D_{p,n}^A$ where L_n are constants and each $D_{p,n}^A$ is a function of values u_s of the parameter in a collection of grid points, called stencil S_n, with individual computation functions, which depend on the numerical framework in which $D_{p,n}^A$ is expressed, and wherein in the approximation $D_{p,n}^A$, the computation functions are chosen in such a way that the approximated value $D_{p,n}^A$ is optimized for the preferential direction.

33. (original) The method of claim 1, wherein $D_p^A = \sum_n L_n D_{p,n}^A$ where L_n are limiting functions of the values u_s of the stencil S, and at least one $D_{p,n}^A$ is a function of values u_s of the parameter in a collection of grid points, called stencil S_n, with individual computation functions, which depend on the numerical framework in which $D_{p,n}^A$ is expressed, and wherein in the approximation $D_{p,n}^A$, the computation functions are chosen in such a way that said approximated value $D_{p,n}^A$ is optimized for the preferential direction.

34. (original) The method of claim 1, wherein the stencil S is chosen from the group consisting of: upwind

discretization stencils, centered discretization stencils, and discretization stencils which are a combination of at least one upwind discretization stencil and at least one centered discretization stencil.

35. (original) The method of claim 1, wherein the numerical discretization is a non-linear discretization.

36. (original) The method of claim 1, wherein the numerical scheme is chosen from the group consisting of: the Lax-Wendroff scheme, the Lax-Friedrich scheme, the MacCormack scheme, the leap-frog scheme, the Crank-Nicholson scheme, the Stone-Brian scheme, the box scheme, Henn's scheme, the QUICK scheme, the κscheme, the Flux Corrected Transport (FTC) scheme, the family of ENO schemes, schemes in the class of the Piecewise Parabolic Method (PPM), multi-level schemes, and schemes obtained with the fractional step method and variants thereof.

37. (original) The method of claim 1, wherein the numerical scheme includes the discretization of a plurality of equations.

38. (original) The method of claim 1 for the numerical simulation of physical phenomena which are modeled by the Navier-Stokes equations with equation(s) of state.

39. (original) The method of claim 1 for the numerical simulation of physical phenomena which are modeled by the Euler equations with equation(s) of state.

40. (original) The method of claim 1 for the numerical simulation of physical phenomena which are modeled by the magneto-hydrodynamic equations with equation(s) of state.

41. (original) The method of claim 1 in combination with at least one model for the physical phenomena chosen from the group consisting of: equations to model turbulence, equations to model chemical reactions, equations to model electromagnetic phenomena, equations to model multiphase flow and equations to model multi-physics phenomena.

42. (original) The method of claim 1 for the numerical simulation of physical phenomena in combination with at least one acceleration technique chosen from the group consisting of: local time-stepping, multi-grid, GMRES and preconditioning.

43. (original) The method of claim 1 for the simulation of a physical phenomenon which includes a material flow.

44. (original) The method of claim 1 for the simulation of a physical phenomenon which includes a material object interacting with material flow.

45. (currently amended) The method of claim 1, wherein the object is ~~for the simulation of a physical phenomenon which includes~~ a vehicle.

46. (currently amended) The method of claim 1, wherein the object ~~for the simulation of a physical phenomenon which~~ includes a rotating blade.

47. (currently amended) The method of claim 1, wherein the object is ~~for the simulation of~~ at least a part of the atmosphere of the earth.

48. (currently amended) The method of claim 1, wherein the flow includes oil ~~for the numerical simulation of~~ .

49. (currently amended) The method of claim 1, wherein the flow includes ~~for the numerical simulation of physical phenomena which include~~ combustion.

50. (currently amended) A data processing system programmed to implement a ~~simulation~~ method according to claim 1.

51. (original) A computer program that can be loaded in a data processing system so as to implement a method according to claim 1.

52. (currently amended) A digital storage computer-readable medium containing a stored computer program that is configured to be loadable in a data processing system so as to control the data processing system to implement a method according to claim 1.